A CATEDRAL, O BAZAR E O CONDOMÍNIO: UM ENSAIO SOBRE O MODELO DE NEGÓCIO DO *SOFTWARE* LIVRE

ENSAIO – TECNOLOGIA DA INFORMAÇÃO

Recebido em: 22/05/2007

Aprovado em: 28/02/2008

Guilherme Finkelfarb Lichand
Mestrando em Economia na PUC-RJ
E-mail: glichand@colband.com.br

Eduardo H. Diniz
Doutor em Administração de Empresas pela EAESP-FGV
Professor da EAESP-FGV
E-mail: eduardo.diniz@fgv.br

Tania Pereira Christopoulos
Doutora pela EAESP da Fundação Getúlio Vargas
E-mail: tchristo@osite.com.br

RESUMO

Este ensaio tem como objetivo apresentar o *software* livre como objeto viável de estratégia de comercialização no mercado de *software*. A partir da análise de seus efeitos sobre as receitas e o *market share* das firmas dominantes, o F/LOSS (*Free/Libre Open Source Software*) emerge como alternativa de modelo de negócio nos segmentos em que o *software* livre apresenta vantagens competitivas sobre o *software* proprietário. Descrevemos as potencialidades e limitações, os aspectos tecnológicos, econômicos e sociológicos da organização da produção de empresas que comercializam produtos ou serviços relacionados ao F/LOSS.

Utilizando-se a perspectiva da Economia dos Custos de Transação e da Economia da Informação, apresenta-se o modelo híbrido de desenvolvimento do F/LOSS nos mesmos moldes de um condomínio – em que a administração central coordena a ação coletiva, enquanto a comunidade, ao mesmo tempo em que integra o processo, monitora as ações da direção centralizada –, como estratégia competitiva alternativa aos modelos de desenvolvimento aberto (Bazar) e proprietário (Catedral).

Palavras-chave: *Software* Livre, *Open Source,* Modelo de Negócios, F/LOSS.

THE CATHEDRAL, THE BAZAAR AND THE CONDOMINIUM: AN ESSAY ON A BUSINESS MODEL OF FREE OPEN SOURCE SOFTWARE

ABSTRACT

Free Open Source Software is described as viable software for commercialization. Considering the revenues and market shares of the dominant suppliers, this software emerges as an alternative business model for segments in which open source software presents competitive advantages over the traditional model. Potentials, limitations and technological, economic and sociological aspects of organizational structures of companies that produce products and services related to Free Open Source Software are described. From the transaction cost and information economy perspectives, a hybrid model to develop Free Open Source Software is presented which is structured according to the concept of a condominium. That is, a central administration coordinates collective action, while the community works and monitors the action of the central direction. This model may be considered as an alternative competitive strategy to peer production (Bazaar) and centralized (Cathedral) models.

Key words*: Free Software, Open-Source Software, Business model, F/LOSS.*

1. INTRODUÇÃO

A conceituação de *software* livre como não comercial, ainda que amplamente difundida, tem sido gradualmente revertida num período curto de tempo. Corroboram esse resultado movimentos como a substituição do termo *free* pelo francês *libre* – o que explicita o caráter libertário do termo em detrimento da ilusória gratuidade do produto – e a aderência de algumas firmas ao *open source software* como parte integrada de seu modelo de negócio.

Entretanto, permanece difusa a possibilidade de adotar o F/LOSS como objeto de estratégia competitiva, em especial por suas implicações econômicas, pouco exploradas no universo multifacetado de contratação de *software*. Adicionalmente, a questão relativa à estrutura organizacional que melhor se adapta aos atributos tecnológicos, comerciais e sociológicos de um modelo de negócio pautado no F/LOSS carece de abordagem mais aprofundada na literatura que avança rapidamente sobre esse fenômeno, tido por alguns autores como uma ruptura do entendimento das fronteiras da firma na análise econômica.

Portanto, este artigo tem como objetivo apresentar o *software* livre como objeto viável de estratégia de comercialização no mercado de *software*, demonstrando que o F/LOSS (*Free/Libre Open Source Software*) emerge como alternativa de modelo de negócio nos segmentos em que o *software* livre apresenta vantagens competitivas sobre o *software* proprietário.

Um debate recorrente na literatura sobre essa tecnologia refere-se à sua denominação: *free software* ou *open source software*. Se, de um lado, a primeira forma é utilizada por alguns autores com uma conotação político-ideológica, em resposta ao oligopólio das gigantes produtoras de *software*, de outro a segunda expressão estaria relacionada exclusivamente ao seu significado econômico e aos aspectos tecnológicos dessa produção.

Como nenhum dos dois fatores pode ser desconsiderado sem a adoção de um viés ideológico, utilizaremos de forma abrangente o termo *Free/Libre Open Source Software* (F/LOSS) ao longo do presente estudo. Conforme discute Cezar Taurion (2004), a abordagem pretendida considera o *software* livre como modelo de negócio, em detrimento de "tecnicismos ou passionalismos".

Nas próximas seções apresentamos uma discussão detalhada à luz da teoria econômica sobre as implicações da ascensão do F/LOSS no mercado de *software* e sobre as particularidades associadas a esse modelo de negócio.

Na seção 2 são detalhadas as dimensões básicas do *software* livre e explicitados alguns conceitos tecnológicos e definições de licenciamento; na seção 3 abordamos a questão da viabilidade do *software* livre como modelo de negócio, avaliando as implicações da adoção dessa estratégia de produção; na seção 4 a discussão recai sobre os modelos de desenvolvimento de *software*: aberto, proprietário e híbrido, com especial ênfase nos aspectos relativos à Economia dos Custos de Transação e à Economia da Informação. Com relação ao modelo híbrido de desenvolvimento, discorremos sobre suas potencialidades e limitações em relação às demais estratégias competitivas; seguem-se as considerações finais, na seção 5.

2. CONCEITOS ECONÔMICOS E TECNOLÓGICOS DO *OPEN SOURCE*

2.1. *Software* Livre e *Software* Proprietário

Software Livre é o *software* cujo código-fonte pode ser livremente acessado. O código-fonte é uma linguagem de computação escrita pelos programadores, com os comandos que determinam o funcionamento dos aplicativos. Quando um programa é compilado para se tornar um programa em funcionamento, esse código é embaralhado e se torna ilegível.

O modelo dominante de comercialização de *software*, chamado de *software* proprietário, é dito fechado, uma vez que é distribuído apenas em código binário, legível somente pelos desenvolvedores da empresa proprietária. A esta também é reservado o direito de proibir ou liberar seu uso, cópia ou redistribuição, de acordo com seu interesse e práticas comerciais.

O modelo proprietário surgiu quando se desenvolveu a indústria de TI e os *softwares* adquiriram valor de comercialização. No início do desenvolvimento da indústria de informática o valor real estava nas próprias máquinas, porque havia poucos computadores, e não nos programas, de maneira que os *softwares* eram gratuitos e livremente distribuídos em formato fonte. Quando

os *softwares* passaram a constituir indústria separada do *hardware*, emergiu um campo de negócios bilionário e as empresas começaram a buscar mecanismos de proteção da propriedade intelectual. Constituindo o próprio conhecimento do programa, o código-fonte passou a ser protegido. Assim, o *software* proprietário surgiu para preencher uma necessidade legítima do mercado (TAURION, 2004).

Um programa é *software* livre se os usuários são livres para acessar seu código-fonte, alterá-lo e redistribuir cópias, com ou sem modificações, de graça ou cobrando uma taxa pela distribuição, para qualquer um em qualquer lugar[1]. Ser livre para fazê-lo significa, entre outras coisas, que não é necessário pedir ou pagar pela permissão.

É importante destacar que *software* livre não significa de domínio público, mas sujeito a licenciamentos que, em maior ou menor grau, permitem a liberdade de usar, copiar, modificar e redistribuir o programa.

"*Software* Livre" não significa "não comercial" ou gratuito. Um programa livre deve estar disponível para uso comercial, desenvolvimento comercial e distribuição comercial, normalmente com custos reduzidos, uma vez que estes são diluídos entre todos que contribuem com a comunidade do *Software* Livre, não havendo necessidade de recuperação desse custo por meio da licença de uso.

A questão central do F/LOSS relaciona-se à sua dinâmica de desenvolvimento. A formulação de *software* livre relaciona-se, tradicionalmente, a comunidades de desenvolvedores; sendo assim, um serviço de desenvolvimento pode ter acesso aos diversos módulos disponibilizados com código aberto e executar as implementações necessárias. Ao final do serviço, o novo código é devolvido à comunidade de desenvolvedores e aos usuários.

[1] Os termos-chave de licenciamento essenciais para definir um código como *open source* são (i) livre distribuição, (ii) código legível e modificável, (iii) permissão de desenvolvimento derivado e (iv) não responsabilização dos desenvolvedores por quaisquer danos advindos da utilização do código licenciado (FROST, 2005).

2.2. Catedral e Bazar

No artigo cujo título dá nome a esta seção, Eric Steven Raymond (2000) distingue dois modelos de desenvolvimento de *software*: o modelo proprietário, cujo processo produtivo, isolado e centralizador se dá nos mesmos moldes da construção de uma catedral, e o modelo de desenvolvimento aberto, que reúne diversas agendas e abordagens e aceita contribuições indiscriminadas, da mesma forma que o mecanismo de constituição de um bazar.

Ao contrário da formalidade e do planejamento presentes no modelo catedral, que tem definição de implantação do *software* em fases, o bazar não prevê planejamento da execução; não há etapa de desenho e este só fica evidente para a comunidade depois da elaboração de algumas versões.

Outras características próprias do bazar são a presença de lideranças que emergem naturalmente, a revisão e a seleção dos códigos feita pelos próprios pares, o ritmo de desenvolvimento dado pela disponibilidade de tempo e a contribuição voluntária dos desenvolvedores.

Para compensar as incertezas do modelo de contribuição voluntária, as comunidades desenvolvedoras adotam alguns mecanismos como a presença de um mantenedor, que é o líder; direcionamento de tarefas mais complexas aos desenvolvedores mais experientes; controle de versões por meio de *softwares* apropriados, a fim de verificar quais versões estão prontas para a apresentação; modularidade, a fim de permitir o trabalho simultâneo de vários colaboradores; e definição de regras claras de codificação, para garantir homogeneidade no código.

2.3. *Open source* x *Software* proprietário

A força de expansão do *software* livre no panorama recente fomenta muitas vezes a idéia de que o F/LOSS substituirá por completo o *software* proprietário. O modelo tradicional de *software* tem seu desenvolvimento muitas vezes pautado pela estratégia do fornecedor, que constantemente cria novas funcionalidades, em geral dessintonizado das necessidades dos usuários, além de um processo de "obsolescência programada" (TAURION, 2004) e uma possível incompatibilidade das novas versões com as anteriores, impedindo a aquisição de versões

anteriores, ainda que atendam perfeitamente às necessidades do usuário.

Todavia, a opção pelo *software* livre fica limitada pelo aprisionamento gerado pelos elevados custos de troca dos *softwares* proprietários, seja em razão da compatibilidade com a base de dados estabelecida, seja pela necessidade de retreinamento do pessoal.

Outro fator que inibe a adoção do *Software* Livre é que muitos *softwares* são de domínio de conhecimento restrito ou envolvem questões estratégicas. Nesse sentido, na visão de Taurion (2004) e Whang (1992), empresas monopolistas ou líderes em seus mercados tendem a adotar padrões proprietários, no intuito de manter sua posição. Em contrapartida, empresas que estejam em busca de competitividade e renovação constante tendem a optar pelo *software* livre, com maior flexibilidade para ajustes aos processos.

2.4. Modelo de licença e modelo de negócio

A prática do F/LOSS engloba duas dimensões distintas: modelo de negócio e modelo de licenciamento, este último com implicações relevantes na esfera econômica.

Há diversas modalidades de licença. Algumas incorporam o princípio da reciprocidade – que estabelece que qualquer desenvolvimento a partir de um código licenciado nesses termos deve, quando o código for distribuído, empregar as mesmas definições legais do código original –, mas outras não.

As mais expressivas licenças de cada grupo são, respectivamente, a *GNU Public License* (GPL) e a *Berkeley Software Distribution License* (BSD), segundo Fink (2003). Se, por um lado, a BSD é tida como modelo 'liberal' de licenciamento – uma vez que permite tanto que qualquer peça de código derivada de um trabalho fundamentado nessa licença seja utilizada sem quaisquer obrigações adicionais, quanto a liberdade do usuário de exibir ou não o código-fonte quando este for distribuído, abrindo espaço para a sua comercialização –, por outro, a GPL estipula que qualquer desenvolvimento subseqüente deve ser licenciado nos mesmos termos de acessibilidade do código e de liberdade de distribuição da peça original.

Há ainda a possibilidade de licenciamento duplo, que se caracteriza por permitir que um produto tenha duas licenças, uma livre e outra proprietária, proporcionando flexibilidade a modelos de negócios que estejam baseados não apenas na utilização, mas também nas receitas oriundas da venda de licenças proprietárias.

Evidentemente, o tipo de licença interfere no modelo de comercialização do *software* e nos seus efeitos econômicos. Analisaremos em mais detalhes essas implicações na discussão sobre a interação entre o modelo de negócio do F/LOSS e a estrutura de mercado.

3. ANÁLISE DE VIABILIDADE DO *SOFTWARE* LIVRE COMO ESTRATÉGIA DE NEGÓCIO

3.1. Natureza das transações

O artigo *Economia em Comunidades* (DINIZ, WILNER e CHRISTOPOULOS, 2005) comenta a ascensão de um fenômeno econômico: "grupos organizados que se comunicam à distância pela Internet com o objetivo de trocar informações, partilhar conhecimentos e realizar tarefas em comum apresentam-se hoje como uma nova proposta de produção econômica". As comunidades de desenvolvedores de *software* são grupos reunidos em torno de um empreendimento comum, que consiste em entregar um produto ou realizar uma tarefa por meio da troca de informações e conhecimentos.

O fator determinante para o trabalho em comunidade é a possibilidade de colaboração fragmentada ou individualizada, por meio da qual cada um contribui com partes específicas para a conclusão dos projetos.

Esse fenômeno é descrito por Benkler (2002) como um campo de estudo absolutamente novo na Economia: uma estrutura produtiva que emerge independentemente de mercados ou de uma base hierárquico-empresarial – diferente, equivalentemente, da abordagem de forma híbrida apresentada pela Economia dos Custos de Transação. O pensamento tradicional sobre o comportamento econômico resistiria à idéia de que milhares de voluntários poderiam colaborar em um complexo projeto econômico e, ainda mais impressionante, ultrapassar as maiores e melhores financiadas empresas do mundo na sua própria área de atuação.

Segundo a teoria de Ronald Coase sobre os custos de transação e a emergência da firma (COASE, 1937 *apud* BENKLER, 2002)[2], os agentes recorrem aos mercados quando os ganhos de agir dessa maneira, sujeitos aos custos de transação, superam os ganhos de fazer o mesmo numa firma, submetidos aos custos de organização. A firma emerge quando o oposto é verdadeiro. Qualquer firma individual estagnará quando seus custos de organização excederem os custos de organização de uma firma menor.

Entretanto, a ascensão do F/LOSS como força substancial no mundo do desenvolvimento de *softwares* coloca em xeque o entendimento tradicional da teoria das organizações: segundo essa interpretação, F/LOSS não se baseia nem em mercados nem em hierarquias empresarias para organizar sua produção. Programadores, usualmente, participam de um projeto não porque assim foram instruídos por um superior nem porque alguém lhes oferece um preço. A massa crítica de participantes em projetos não pode ser explicada pela presença direta de um comando, um preço ou mesmo um retorno monetário futuro, sobretudo com relação à importante decisão quanto aos projetos de que esses indivíduos tomam parte.

Segundo Watson *et al*. (2005), são os benefícios da transação (benefícios oriundos de transações econômicas) que motivam os membros a permanecerem contribuindo com as comunidades e que atraem novos talentos. Esses benefícios vão além dos benefícios financeiros comuns aos modelos da Firma e do Mercado. Entre esses benefícios, destacam-se a reputação e o desenvolvimento intelectual e de habilidades. A existência e validade desses benefícios devem ser consideradas relativamente à cultura, às características individuais, às necessidades e à tecnologia de informação disponível (WATSON *et al*., 2005). Em relação a este último aspecto, os participantes são atraídos por projetos que propiciam o desenvolvimento de novas habilidades. Assim, projetos baseados em novas tecnologias tendem a atrair maior número de participantes.

A teoria da Economia dos Custos de Transação (ECT) foi selecionada entre outras teorias econômicas porque adota a transação como unidade de análise e não simplesmente as *commodities*, como fazem outras teorias econômicas, para compreender a eficiência dos modelos de negócios. A eficiência é analisada com base nas estruturas de governança adotadas e nos respectivos custos de transação gerados. Tratando-se do *software* livre, o que nos interessa é o custo da transação e, mais especificamente, das estruturas de governança adotadas.

Na ECT são abordadas três estruturas de governança: mercado, formas híbridas e hierarquia. A primeira e última formas contrapõem-se no *trade-off* entre incentivo e controle. Se, por um lado, o mercado apresenta incentivos fortes, associando esforço a remuneração e exercendo pouco controle sobre essas transações, essa associação se dá com menos intensidade na firma, que, por outro lado, promove incentivos tênues a um comportamento cooperativo e adota controles internos capazes de sobrepujar a ausência de incentivos fortes. As formas híbridas são analisadas como estruturas que apresentam tanto as formas extremas do mercado quanto a hierarquia, ganhando em incentivos e perdendo em controle à medida que se aproximam das primeiras e se afastam da segunda.

Formas produtivas em que grupos de indivíduos colaboram em projetos, estimulados por motivações outras que não a remuneração do mercado ou o comando gerencial, são denominadas *peer production* por Benkler (2002) e podem ser analisadas segundo as propriedades das formas híbridas de estrutura de governança, o que conduz a implicações conclusivas para as potencialidades e limitações da adoção desse modelo produtivo como estratégia competitiva.

3.2. Efeitos sobre a estrutura de mercado

Os efeitos da ascensão do *open source* no mercado de *software*, em contraposição ao modelo proprietário, foram esboçados por Frost (2005), em termos de análise econômica, relativamente à receita e ao *market share*. Mercados de tecnologia apresentam por vezes elevadas barreiras à entrada, em virtude: 1) da necessidade de investimentos tidos como custos irrecuperáveis, do aprisionamento dos usuários – derivado dos custos de troca inerentes à base de dados associada à tecnologia operante e da necessidade de retreinamento de pessoal e de custos de aquisição e renovação de licença –, e 2) da presença de externalidades de

[2] COASE, R. The Nature of the Firm. *British Academic Journal Economica*, v. 4, n. 16, Nov. 1937.

rede, que consistem em *payoffs* crescentes associados à crescente utilização de uma diretriz tecnológica. Essas duas variáveis fornecem aos padrões de mercado uma base de autoconservação.

Comunidades organizadas em torno da *peer production* deparam-se com reduzidos custos irrecuperáveis, dados o perfil não monetário de incentivos dos programadores e a diluição dos custos entre eles. Associada a um desenvolvimento potencialmente mais rápido dos códigos a partir da confluência de múltiplos e simultâneos processos criativos, essa configuração estaria vinculada a um processo mais acelerado de desvalorização (FINK, 2003), o que postaria um desafio às receitas e ao *market share* das firmas preestabelecidas.

Como vantagem adicional, o *software* livre apresenta a propriedade fundamental de redução da especificidade de ativos – dimensão que atrela o valor de um ativo determinado à continuidade de uma relação à qual ele é específico. Assim, ativos específicos são aqueles não reempregáveis sem perda de valor, o que, associado ao oportunismo e à incompletude dos contratos, imputa custos de transação ao investimento nesses ativos. Essa característica expressa que para a firma contratante do serviço de desenvolvimento de *software* os custos de troca são menores, uma vez que não há custos de renovação de licença (VARIAN e SHAPIRO, 1999).

Em uma dimensão mais abrangente, a especificidade de ativos tem impacto sobre a estrutura de governança (WILLIAMSON, 1985). À medida que se eleva a especificidade de ativos, mais controle é demandado sobre a transação, de maneira a evitar o comportamento oportunista, potencial às partes envolvidas. Com isso em mente, tanto mais próxima da hierarquia será a estrutura ótima quanto maior a especificidade de ativos da relação envolvida. Nesse sentido, a redução da especificidade de ativos inerente à inserção do *open source* no mercado de *software* pode também ter efeitos reversos sobre a concentração nesse mercado.

Por fim, a aceleração do processo de desvalorização põe em cheque a política de "obsolescência programada", própria à indústria de *software*, e desafia o modelo de negócio tradicional desse mercado.

Com efeitos inegáveis sobre as receitas e o *market share* das firmas dominantes, o F/LOSS é uma alternativa de estratégia competitiva nesse mercado. Analisaremos na próxima seção suas potencialidades e limitações como modelo de negócio.

3.3. Firmas de F/LOSS e estratégia competitiva

A questão central no debate acerca de um modelo de negócio para o F/LOSS é a capacidade de auferir lucros a partir da adoção dessa estratégia competitiva.

Uma esfera de atuação proeminente e que vem sendo incorporada mesmo por grandes corporações ícones do modelo proprietário, tais quais a IBM, é o uso do *open source software* como plataforma de desenvolvimento (FROST, 2005). Essas empresas atuam desenvolvendo soluções (proprietárias ou não) para necessidades de negócio específicas de cada empresa, expressas por meio de avanços sobre o código original ou adaptações às particularidades de demanda.

As licenças têm um papel importante na comercialização do *Software* Livre, pois muitas, dentre elas a BSD, abrem espaço para a comercialização e distribuição dos avanços ou implementações do produto original sem disponibilização do código-fonte, em sintonia com as necessidades comerciais de uma firma baseada no modelo *open source*. Mesmo a GLP pode conviver lado a lado com o código licenciado (FINK, 2003), ainda que careça de jurisprudência (FROST, 2005).

A adoção desse modelo de negócio e a redução da especificidade de ativos intrínseca à dinâmica do F/LOSS tendem a conduzir a estruturas de mercado mais competitivas ao diminuírem a rigidez das relações contratuais.

4. MODELOS DE DESENVOLVIMENTO DE *SOFTWARE*

4.1. Modelo de desenvolvimento aberto

Nesta seção, analisaremos a situação em que a firma se insere como mais um elemento em uma comunidade de prática virtual, integrando a forma produtiva do *peer production*, em condição equivalente a qualquer um dos demais

desenvolvedores. Analisamos a decisão de contratação de um desenvolvedor por parte de um consumidor (sob a figura de indivíduo, organização ou Estado) e as relações envolvidas no processo produtivo.

A literatura da ECT aborda amplamente a construção de modelos teóricos, no intuito de comparar a eficiência relativa das diversas estruturas de governança em cada tipo de transação. O modelo de Williamson (1985) aqui apresentado como uma ilustração à discussão subseqüente é retirado de Azevedo (1996): "apresenta a essência dos argumentos da ECT em uma forma analítica reduzida, em que a variável-chave é a especificidade dos ativos[3]. A escolha da forma organizacional reflete, sobretudo, essa dimensão das transações. As demais dimensões – incerteza e freqüência – e elementos do ambiente institucional – como garantia de direitos de propriedade, disponibilidade de informações, códigos de ética, etc. – representam parâmetros de deslocamento das funções da forma analítica reduzida do modelo".

A distinção mais fundamental se refere ao contraponto entre a descentralização do mercado, em que partes autônomas transacionam, e a centralização da firma, em que o proprietário conserva o poder de *fiat* ("faça-se"), resolvendo desavenças contratuais internamente, sem necessidade de recorrer a instâncias judiciais. Ainda que essa necessidade exista no caso do mercado, as partes são autônomas para negociar.

A capacidade de adaptação definirá o critério fundamental para avaliar as estruturas de governança ótimas para cada transação específica, dado que essa problemática se reflete na maior parte dos custos de transação. "Nesse ponto entra em cena a especificidade dos ativos, em cuja presença verifica-se uma dependência bilateral e, portanto, a necessidade de uma adaptação do tipo cooperativa" (AZEVEDO, 1996).

O mercado, por gerenciar informações com menos custos e por apresentar incentivos de alto poder, tem maior capacidade de adaptação na resolução de problemas autônomos. Alternativamente, a hierarquia apresenta vantagem na emergência de problemas cooperativos, em que há dependência multilateral.

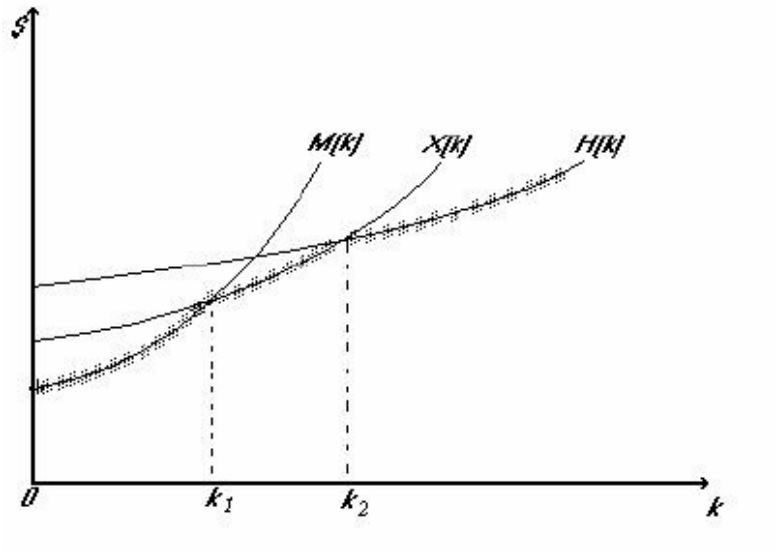
A incerteza, dimensão das transações introduzida anteriormente, precisa ser compreendida de forma mais depurada em razão da conceituação diferenciada das comunidades de desenvolvedores de *software* como formas híbridas.

A importância da incerteza na definição dos mecanismos de governança refere-se à suscetibilidade da capacidade de adaptação a cada um dos problemas analisados previamente à variação dos eventos. O ponto é que as estruturas intermediárias tendem a ser mais sensíveis aos efeitos da incerteza, em razão dos custos de renegociação. Assim, na iminência de uma querela contratual, diante de uma elevação da incerteza, que acentua as lacunas *ex ante* à transação, os custos de renegociação do mercado sobem, mas não significativamente, dado que as partes são autônomas para negociar. Há que considerar aqui que se está tratando de uma análise econômica da forma híbrida sem a possibilidade de fixação de regras *ex ante*. Na seção 4.3 serão apresentadas evoluções do modelo, que minimizam a incerteza entre as partes.

Do ponto de vista da hierarquia, em razão da existência do poder de *fiat* decorrente do controle central, uma situação nos mesmos moldes que a apresentada no modelo aberto também eleva de forma pouco significativa os custos de renegociação. Já quanto às formas híbridas, as partes são interdependentes, o que resulta em uma elevação substancial dos custos de renegociação diante de um cenário de incerteza, quando não são regulamentadas as transações, exigindo consenso bilateral.

[3] Seis tipos de especificidade de ativos são identificados por Williamson (1985): i) especificidade locacional – a localização próxima de firmas de uma mesma cadeia produtiva economiza os custos de transporte e armazenagem e significa retornos específicos a essas unidades produtivas; ii) especificidade de ativos físicos; iii) especificidades de ativos humanos, ou seja, toda forma de capital humano específico a uma determinada atividade; iv) ativos dedicados – relativos a um montante de investimento cujo retorno depende da transação com um agente particular e, portanto, relevante individualmente; v) especificidade de marca, que se refere ao capital – nem físico nem humano – que se materializa na marca de uma empresa, sendo particularmente relevante no mundo das franquias; e f) especificidade temporal, onde o valor de uma transação depende sobretudo do tempo em que ela se processa, sendo especialmente relevante no caso da negociação de produtos perecíveis.

Gráfico 1: Especificidade de ativos e custos de transação associados às estruturas de governança



Fonte: AZEVEDO (1996).

O gráfico anterior remete às três estruturas de governança exploradas em detalhe, identificando os custos de transação associados a cada nível de especificidade de ativos (k) – M correspondendo à curva de mercado, que apresenta custos de transação mais reduzidos se a especificidade de ativos é nula, e custos que se elevam mais rapidamente quando cresce k, comparativamente às demais formas organizacionais (primeira derivada mais elevada, M'); H representa hierarquia, indicada por comportamento inverso, uma vez que apresenta os mais elevados custos de transação se a especificidade é nula, e o menor incremento (H') à medida que k aumenta; X representa de maneira abrangente as formas híbridas de comportamento intermediário entre as estruturas extremas.

Embora não explicitado no modelo, Williamson sugere que as mudanças nos parâmetros de deslocamento afetam de modo diferente cada mecanismo de governança. O aumento da incerteza, por exemplo, tenderia a aumentar relativamente mais os custos da forma híbrida, uma vez que, de um lado, adaptações que exigem cooperação – o que não acontece com a forma de mercado – estariam mais sujeitas a atitudes oportunistas, e, de outro, os problemas impostos pelas contingências necessitam de uma solução consensual – o que não acontece na forma hierárquica (AZEVEDO, 1996).

Esse efeito é explicitado mediante um *shift up* em todas as curvas (mercado, hierarquia e formas híbridas), com maior intensidade para esta última estrutura. Em última instância, o resultado amplia a área das formas extremas e reduz a da estrutura intermediária à medida que aumenta o nível de incerteza. Altera-se, subseqüentemente, o nível de especificidade de ativos que determina para que grau de especificidade das transações cada uma das formas configura estrutura ótima de governança (com uma nova curva envelope).

Sob esse aspecto, a incerteza configura elemento determinante para a escolha da forma organizacional. Adicionalmente, a conceituação de incerteza associada à assimetria informacional também acrescenta à discussão a diferença entre resultados observáveis e resultados verificáveis. Se parte ou todos os participantes de uma transação desconhecem o parâmetro de avaliação ou monitoramento, isso impede a construção de esquemas de incentivos adequados (AZEVEDO, 1996).

4.2. Modelo de desenvolvimento proprietário

Nesta seção analisamos as implicações econômicas da estrutura de governança própria da opção do modelo de desenvolvimento fechado *cathedral-like*.

Há alguns riscos inerentes ao modelo proprietário de *software* que, segundo Watson *et al*. (2008), podem ser minimizados em outros modelos. O primeiro desses riscos é a ofensa a direitos autorais. Uma ofensa a direitos de propriedade de um *software* proprietário só será descoberta muito tempo depois, podendo causar danos que não ocorrem em estruturas de mercado em que o código-fonte é aberto e disponível para todos e no qual uma ofensa a direitos de propriedade poderia ser mais facilmente descoberta (WATSON *et al*., 2008). A impossibilidade relativa de rastreabilidade do código do *software* alterado eleva os custos do modelo, pois as corporações que o desenvolvem e comercializam devem prever em seus preços perdas relativas às conseqüências de ofensas a direitos autorais.

Esse risco pode ser minimizado com uma definição contratual de eventuais indenizações ao contratante. A estrutura geral de contratos de *software* usualmente consiste em três partes: definição do produto, proteção de propriedade intelectual e estrutura de pagamento. Esses aspectos podem ser respaldados por definições estabelecidas na Lei brasileira nº 9.609/98. Entretanto, por se tratarem de aspectos contratuais, nem sempre são definidos de acordo com o que a lei define. Assim, com relação à proteção da propriedade intelectual, como os contratos requerem especificação das definições pelas partes, há sempre um campo potencial de conflito acerca da pertença ou licença da propriedade intelectual do ativo desenvolvido, e o risco de que informações estratégicas de uma das partes sejam reveladas à outra.

Quanto ao pagamento, grande parte dos contratos especifica os pagamentos segundo o licenciamento, serviço, treinamento, manutenção e documentação e/ou segundo os módulos do projeto, mas a especificação de um plano contingencial em caso de fracasso depende de definições prévias das partes, o que nem sempre ocorre. Para reduzir as incertezas relativas, todo contrato deve, portanto, definir explicitamente um plano contingencial (WHANG, 1992), garantindo a parte do contratante e também a do fornecedor.

A partir dos riscos analisados, pôde-se verificar que maior incerteza é um parâmetro associado à construção de incentivos adequados para a construção de *software* no modelo proprietário, quando comparado ao modelo de desenvolvimento aberto.

Apesar de os custos relativos à incerteza poderem ser minimizados em razão de sua previsibilidade no contrato, há outros custos relevantes que oneram o modelo proprietário, comparativamente a outros modelos. O primeiro é o custo da infra-estrutura necessária para desenvolvimento e distribuição do *software*. Na indústria tradicional de *software*, que adota o modelo proprietário, o desenvolvimento do *software* ainda exige a presença física dos desenvolvedores, elevando custos de infra-estrutura comparativamente a outros modelos de desenvolvimento remoto de *software*, como em uma estrutura de mercado, por exemplo, em que cada um pode trabalhar desenvolvendo o código em seu próprio ambiente.

Na próxima seção investigaremos uma configuração intermediária capaz de fornecer os incentivos adequados a ambas as partes envolvidas na relação contratual e que corresponde a um modelo de negócio alternativo aos demais modelos.

4.3. Análise do modelo de desenvolvimento híbrido

4.3.1. Justificativa econômica

Nos tópicos anteriores, vimos que custos de renegociação elevados associados ao desenvolvimento segundo o modelo aberto e elevados custos de monitoramento, próprios de ambos os modelos de negócio, emergem como limitações demarcadas da estratégia competitiva delineada pela adoção dessas estruturas organizacionais.

Um modelo de negócio que vinculasse seu processo produtivo a uma comunidade de prática virtual, organizada sob o modelo *peer production*, não como um desenvolvedor adicional, em condição equivalente a todos os demais, mas na posição de coordenador do desenvolvimento contratado, responsável por todas as implicações legais e financeiras do projeto, incorporando seus membros na sua execução, seria uma alternativa aos modelos anteriores.

Nesse contexto, a responsabilidade de desenvolvimento recai sobre uma gestão centralizada, menos sensível à incerteza e com margem para renegociação, ao mesmo tempo em

que os desenvolvedores pulverizados no sistema de *peer production* devem ser capazes de prover monitoramento do desenvolvimento do projeto.

Ao mesmo tempo, conserva-se a flexibilidade da estrutura de governança – por meio da interação com a comunidade de desenvolvedores –, relevante diante de outras formas de incerteza, como a imprevisibilidade tecnológica (HARRIGAN, 1987 *apud* AZEVEDO e ROCHA, 2000).

Adicionalmente, o risco do projeto é diluído entre a corporação e a comunidade de desenvolvimento de F/LOSS, minimizando o problema da degeneração do mercado, também segundo essa perspectiva. É a estrutura não monetária de *payoffs* dos desenvolvedores – guiados pela busca de reputação, aprendizado ou contribuição para um grande projeto – que possibilita essa interação híbrida entre as culturas voluntária e capitalista.

Raymond (2000) associa o modelo de desenvolvimento proprietário ao processo de construção isolado, centralizador, de uma catedral, e o modelo de desenvolvimento aberto, reunindo diversas agendas e abordagens e aceitando contribuições de virtualmente qualquer um, ao mecanismo de constituição de um bazar. O modelo híbrido de desenvolvimento proposto neste artigo pode ser associado a um condomínio, em que a administração central coordena a ação coletiva enquanto a comunidade integra o processo e monitora as ações da direção centralizada.

Em primeiro lugar, cabe ressaltar que esse esquema produtivo é adequado tão-somente a um escopo limitado de projetos de *software*: aqueles que não apresentam restrições a informações estratégicas ou compartilhamento de dados e são relativos a dimensões pouco concentradas do mercado de *software*, em que o modelo tradicional não apresenta vantagens competitivas em razão de ganhos baseados exclusivamente em licenciamento, por exemplo.

Há evidências da viabilidade do modelo de *peer production*, apresentadas no estudo de Benkler (2002): o projeto da *Wikipedia Project* envolve milhares de voluntários que colaboram para escrever uma enciclopédia. O projeto é desenvolvido por meio de um *software* em código aberto, cujo foco é permitir múltiplas colaborações na edição dos documentos. Em 2002, com 18 meses de operação, havia 2 mil pessoas colaborando e 30 mil artigos já produzidos. O que difere a Wikipedia de outros modelos de enciclopédia *on-line* é o caráter autoconsciente dos voluntários que se dedicam ao projeto de forma produtiva e a baixa quantidade de tentativas de deturpar o objetivo do projeto.

Benkler (2002) cita o projeto Kuro5hin, ou K5, como o "estado-da-arte" do *peer production*. O K5 é uma comunidade que, em 2002, abrigava 25 mil colaboradores de artigos nas áreas de cultura, tecnologia, política e comunicações, enfatizando a qualidade do material publicado. O controle de qualidade é executado por um *software* em código aberto, reforçado pelos mecanismos de *peer-review* antes da publicação e *peer-commentary* após a publicação.

Watson *et al*. (2005) corroboram os exemplos de Benkler (2002), argumentando que o sistema de produção que vem sendo definido como próprio do *Open Source* pode ser visto desde há muito tempo em qualquer organização em forma de comunidade, pois esta corresponde a um grupo organizado de pessoas trabalhando coletiva e voluntariamente para alcançar um objetivo comum. Entre vários exemplos, a Wikipedia é citada por Watson *et al*. (2005) como exemplo de organização em comunidade.

Atualmente, verificam-se variações do modelo de comunidades de *software* livre, na tentativa de viabilizá-lo. Segundo Watson *et al*. (2008), há quatro modelos já operacionalizados no mercado de *software* livre:

- Comunidade aberta: o desenvolvimento e suporte do *software* é feito por voluntários. Não há interesse comercial e este modelo parece dominar o mercado, pois há mais de 150 mil projetos na *SourceForge* (2008).
- Distribuição corporativa: empresas identificam produtos com potencial para o Mercado e interagem com a comunidade desenvolvedora, investindo no desenvolvimento de habilidades de serviços de suporte. Exemplos dessas empresas são: RedHat, SpikeSource e OpenOSX, que nasceram criando valor e gerando receita por meio da identificação de projetos interessantes para o mercado e melhorando os métodos de distribuição e de suporte desses produtos.
- OSS patrocinado: corporações e/ou fundações patrocinam projetos de OSS. Os projetos são

muitas vezes iniciados por empresas que liberam o código fechado e encorajam seus funcionários a continuar trabalhando no novo projeto aberto. A Sun Microsystems criou sua comunidade de desenvolvedores para aquisição de informações e colaborações ao seu sistema (DINIZ, WILNER e CHRISTOPOULOS, 2005). A Zope.corporation, por exemplo, estimula o desenvolvimento de uma comunidade de desenvolvedores voluntários de *software* de gestão de conteúdo, com base em um *software* livre, com o objetivo de incrementar seu poder de consultoria no mercado de *softwares*. Apesar de voluntários, esses desenvolvedores estão inseridos numa estrutura de incentivos das empresas-clientes da Zope.corporation, o que é um ponto-chave de diferenciação em relação ao problema enfrentado na contratação direta de desenvolvedores independentes ou de uma corporação organizada nos moldes proprietários.

- Segunda geração: também chamado *Professional Open Source* ou OSSg2. É um híbrido dos modelos Distribuição corporativa e OSS patrocinado. As corporações que distribuem não cobram pelo código, mas controlam seu desenvolvimento e retêm para si o que há de melhor em qualidade de serviços de manutenção desses códigos. Exemplos de empresas de OSSg2 são: JBoss, MySQL, Trolltech e Sleepycat.

Segundo Watson *et al*. (2008), as OSSg2 têm um modelo de negócios que emergirá como dominante para o desenvolvimento de OSS nos próximos anos. Esse modelo aproxima-se do proposto por nós neste tópico – o Condomínio –, pois vincula seu processo produtivo a uma comunidade de prática virtual, organizada sob o modelo *peer production*, no qual a corporação ocupa uma posição de coordenação do desenvolvimento.

Watson *et al*. (2008) acreditam que há diferenças nos modelos de negócios das empresas OSSg2 que merecem ser citadas. A primeira diferença é que algumas são baseadas na estratégia da licença dupla, oferecendo opções de licença OSS ou comercial aos clientes. Esta última opção é para prover clientes que desejam que suas modificações não sejam liberadas para a comunidade. As empresas que adotam a licença dupla acreditam que esse modelo permite competir com os fornecedores de SW proprietários. Outras, no entanto, preferem basear-se apenas na Lesser General Public License (LGPL) – caso da JBoss – e recebem receitas somente de serviços que incluem suporte, treinamento e consultoria.

A segunda diferença é que, enquanto a Trolltech, a MySQL e a Sleepycat são proprietárias de seu código-fonte, a fim de poderem oferecer seu modelo de licença dupla, a JBoss não o é, mas apesar disso controla o código mais rigidamente que outras OSS com outros modelos de licença.

Três importantes características trazem vantagens ao modelo das OSSg2: 1) atribuição clara de responsabilidades, pois, como o código é aberto, alterações e tentativas de "ataque" ao código-fonte podem ser descobertas mais cedo que aquelas feitas aos códigos proprietários e podem ser corrigidas a tempo; 2) base de talentos desenvolvida com menores riscos e custos de contratações erradas, porque os contratados devem ter sido voluntários e demonstrado que são capazes de desenvolver o sistema e dar-lhe manutenção; e 3) ecossistema desenvolvido assegura qualidade e rastreabilidade do *software*. O ecossistema inclui todas as entidades que ganham com a presença das OSSg2 no mercado, como serviços de suporte, autorias, educação, publicidade, sócios e comunidades de usuários. Isso se traduz em múltiplos *web sites*, *e-mails*, listas, *newsgroups*, conferências e outros materiais com informações atualizadas sobre produtos das OSSg2. Assim, o ecossistema provê um efetivo suporte de pré-vendas. Além disso, enquanto o produto é lançado, há milhares de pessoas fazendo testes, volume muito maior que os testadores oficiais dos *softwares* proprietários. Esses benefícios contribuem para a redução de três riscos estratégicos específicos do modelo OSSg2, que devem ser discutidos: demanda, eficiência e inovação.

- Demanda: o risco de baixa demanda, derivado de um modelo relativamente novo de negócios, é compensado porque, com o custo baixo, resultado da eficiência operacional devida às características acima mencionadas, o preço também pode ser baixo, atraindo novos clientes.
- Eficiência: o risco de ausência de eficiência é minimizado pela utilização da Internet como base para transferir os *softwares* aos computadores dos desenvolvedores e dos clientes, reduzindo custos operacionais.
- Inovação: riscos de baixa inovação não existem. Ao contrário, falhas podem ser corrigidas mais

rapidamente e sugestões de mudanças também são feitas por pessoas do mundo todo.

Verifica-se que há uma considerável redução de riscos e custos do modelo híbrido na versão Condomínio, em razão das evoluções nos modelos de negócios que o *Software* Livre vem experimentando nos últimos anos. A ausência de segurança, que parecia ser uma grande desvantagem da adoção do *Software* Livre em um modelo de mercado, por exemplo, revela-se uma vantagem, dados o rápido e fácil acesso ao código-fonte e a coordenação de uma corporação que pode assumir responsabilidades por esse código, respaldada pela reciprocidade na cooperação e coordenação dos desenvolvedores, como é o caso do modelo das OSSg2.

4.3.2. Fundamentos sociológicos

Em Yuwei Lin (2006), uma das entrevistas com um programador de empresa de F/LOSS de porte intermediário traz o seguinte relato: "*I think the company should go closer to the Linux community. We would like to keep some information confidential for business, but we should not forget the open source ideology*". A citação expressa que o modelo do condomínio já é encarado como modo de organização da produção por diversas corporações. O que o presente artigo procura afirmar, alternativamente, é que esta estrutura de governança é preferível às demais formas extremas e, portanto, referenciada como estratégia de comercialização, pode constituir vantagem competitiva no processo de contratação de projetos de desenvolvimento de *software*.

Mais ainda, diversos aspectos da interação entre essas diferentes culturas de gestão do conhecimento e organização da produção corroboram para uma mútua dependência entre comunidades de desenvolvedores de *software* e corporações, como por exemplo a detecção de *bugs* e a contribuição com *patches* por parte da comunidade, muitas vezes pautadas no desejo de inovação dos desenvolvedores (LIN, 2006). É evidente que essa dinâmica é erigida sobre os riscos intrínsecos às relações informais, mas firmas de F/LOSS efetivamente economizam muito tempo e recursos quando vinculadas a comunidades de prática virtuais ligadas a corporações.

Sob a ótica dessa interdependência, Lin (2006) analisa que as diversas licenças de F/LOSS configuram centralmente um mecanismo para que mais atores se engajem no processo de inovação do *open source*, englobando o máximo de desenvolvedores nessas redes sociais, de maneira a potencializar o processo criativo – ainda que algumas licenças, como a GPL, restrinjam sua interação com o *software* proprietário. Assim, as licenças servem para incrementar a colaboração entre firmas de F/LOSS e comunidades de desenvolvedores.

Certamente, entretanto, a interação entre essas diferentes culturas organizacionais não é isenta de conflitos. Na próxima seção, analisaremos limitações relevantes à adoção do modelo do condomínio.

4.4. Novo *trade-off*

Cabe retomar a questão pouco definida de incentivos aos desenvolvedores. Se a ideologia é componente importante da motivação dos programadores que constituem as comunidades de prática virtuais, esses incentivos podem comprometer o envolvimento em projetos coordenados por uma firma pautada por um modelo de comercialização de *software*, ainda que *open source* (STEWART e GOSAIN, 2003).

Adicionalmente, existe a possibilidade de, uma vez vinculado o desenvolvimento à configuração da *peer production*, um grupo assumir o *software* e gerar uma variante diferenciada e mesmo incompatível com o código original (TAURION, 2004). Não obstante, algumas inovações podem não ser incorporadas ao projeto de *software*, em razão do não-compartilhamento de conhecimentos específicos ou idéias inovadoras pela comunidade desenvolvedora, que pode acoplá-los a um novo algoritmo, gerando uma nova oferta no mercado – mormente sob os moldes proprietários.

É importante ressaltar que questões de cultura e política locais e necessidades individuais e da tecnologia de informação disponível para gerar novas habilidades deverão influenciar na agregação de comunidades para a realização de um objetivo comum (WATSON *et al*., 2005).

Entretanto, as questões desagregadoras anteriormente detalhadas podem ser minimizadas com o advento da estrutura produtiva híbrida. Assim como abordado na discussão sobre o caso da Zope.corporation, o controle da gestão por parte de

uma empresa implica vinculação diferenciada a uma comunidade de desenvolvedores de *software* – no caso, composta de desenvolvedores voluntários estimulados a participar pelas próprias empresas em que trabalham, clientes da Zope.corporation. Esse *design* funcional minimiza a questão dos incentivos, providos por uma firma (estrutura integrada verticalmente, em termos da comunidade de desenvolvedores de *software* que corresponde a seus funcionários) que, de acordo com a literatura da ECT, representa o caso extremo do *trade-off* entre controle e incentivo, em contraposição ao mecanismo de mercado.

Equivalentemente, a gestão hierárquica da informação no projeto de *software* permite a diluição de parte da insegurança gerada pela possibilidade de omissão estratégica de conhecimentos ou idéias pelos desenvolvedores organizados sob a *peer production*.

É evidente que isso faz emergir um novo *trade-off*, entre monitoramento e segurança, cuja decisão de balanceamento depende sobretudo das características do *software* contratado, conforme mostra a tabela a seguir.

Tabela 1: *Trade-off* entre monitoramento e segurança nos modelos produtivos de *software*

Modelo de NEGÓCIO	Estrutura de Governança	*Trade-off*
Catedral	Firma nos moldes proprietários	Controle – incentivo
Bazar	Firma como desenvolvedor adicional numa comunidade *peer production*	Controle – incentivo
Condomínio	Firma como coordenadora do desenvolvimento vinculada a uma comunidade *peer production*	Monitoramento – segurança

Fonte: Elaborada pelos autores.

5. CONSIDERAÇÕES FINAIS

Ao longo do presente estudo, apresentamos o *software* livre como objeto viável de estratégia de comercialização no mercado de *software*. Com efeitos inegáveis sobre as receitas e o *market share* das firmas dominantes, o F/LOSS apresenta-se como alternativa de modelo de negócio nesse mercado, nos segmentos em que o *software* livre apresenta vantagens competitivas sobre o *software* proprietário e que não sejam de informação restrita ou estratégica.

Na comparação entre os modelos de desenvolvimento aberto e proprietário, observa-se que tanto um quanto outro apresentam restrições importantes. Vimos que custos de renegociação associados ao desenvolvimento de *software* segundo o modelo aberto, risco elevado de perdas derivadas de ofensas a direitos autorais e elevados custos de monitoramento, inerentes a ambos os modelos de negócio, emergem como limitações demarcadas à estratégia competitiva delineada pela adoção dessas estruturas organizacionais.

Diante disso, um movimento estratégico da corporação consistiria na adoção de um modelo de negócio que vinculasse seu processo produtivo a uma comunidade de prática virtual, organizada sob o modelo *peer production*, não como um desenvolvedor adicional, em condição equivalente a todos os demais, mas na posição de coordenador do desenvolvimento contratado, responsável por todas as implicações legais e financeiras do projeto, incorporando a comunidade à sua execução, possivelmente vinculada a uma organização.

O modelo híbrido de desenvolvimento de F/LOSS nos mesmos moldes do que aqui se convencionou chamar de condomínio – em que a administração central coordena a ação coletiva enquanto a comunidade, ao mesmo tempo em que integra o processo, monitora as ações da direção centralizada –, em detrimento do bazar ou da catedral, apresenta-se como estratégia competitiva superior. Essa avaliação é derivada das vantagens da gestão centralizada, que reduz a sensibilidade à incerteza, ao mesmo tempo em que conserva a flexibilidade da interação com a comunidade de desenvolvedores, relevante diante de outras formas de incerteza, como a imprevisibilidade tecnológica.

Adicionalmente, os desenvolvedores pulverizados no sistema de *peer production* devem ser capazes de prover monitoramento do desenvolvimento do projeto, cujo risco é diluído entre a corporação e a comunidade de desenvolvimento do F/LOSS, minimizando o problema da degeneração do mercado.

Ainda, essa estrutura de governança pode servir a outros propósitos de cunho comercial, como a divulgação de novos produtos ou a provisão de garantias. Não obstante, a interação entre as culturas voluntária e do modelo proprietário está associada a ganhos na detecção de *bugs* e na contribuição para o desenvolvimento de *patches*, e a incentivos próprios à constituição das redes sociais nas comunidades de desenvolvedores de *software* que concorrem favoravelmente aos objetivos da firma adepta do modelo do condomínio.

Por outro lado, existe a possibilidade de comprometimento da participação de desenvolvedores organizados em comunidades em projetos coordenados por firmas com orientação comercial, em razão de componentes ideológicos, ainda que minimizados pela vinculação diferenciada da firma a uma comunidade de prática virtual. Surge ainda um novo *trade-off*, entre segurança e monitoramento.

Por fim, diversos modelos de licenciamento são desenvolvidos de maneira a se adaptarem às especificidades comerciais, tecnológicas ou ideológicas de cada firma, mas ainda assim conservam-se fiéis aos princípios do código aberto. A questão central reside na escolha de um modelo de licenciamento que equilibre a participação da comunidade no processo criativo e as potencialidades de distribuição comercial do *software* contratado.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AZEVEDO, P. F. *Integração vertical e barganha*. Tese (Doutorado em Economia) – Faculdade de Economia, Administração e Contabilidade. São Paulo: Universidade de São Paulo, 1996.

AZEVEDO, P. F.; ROCHA, M. M. Governança ineficiente: uma análise das transações na indústria petroquímica brasileira. *Revista da Anpec*, v. 16, n. 3, p. 127-156, 2000. Disponível em: <www.anpec.org.br>. Acesso em: 18 mar. 2008.

BENKLER, Y. Coase's Penguin, or, Linux and The Nature of the Firm. *The Yale Law Journal*, v. 4, n. 3, Ago. 2002.

DINIZ, E.; WILNER, A.; CHRISTOPOULOS, T. Economia e Comunidade. *GV Executivo*, São Paulo, v. 4, n. 2, p. 16-21, maio-jul. 2005.

FINK, M. *The Business and Economics of Linux and Open Source*. Upper Saddle River, NJ: Prentice Hall PTR, 2003.

FROST, J. F. *et al*. *Some Economic & Legal Aspects of Open Source Software*. University of Washington. Department of Economics, 2005. Disponível em: <http://opensource.mit.edu/papers/frost.pdf>. Acesso em: 20 jan. 2006.

LIN, Y. Hybrid Innovation: How Does the Collaboration Between the FLF/LOSS Community and Corporations Happen? *Knowledge, Technology and Policy Journal*, v. 18, n. 4, 2006.

RAYMOND, E. S. *The Cathedral and the Bazaar*, 2000. Disponível em: <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>. Acesso em: 20 jan. 2006.

STEWART, K.; GOSAIN, S. *The impact of ideology on effectiveness in open source software development teams, 2003*. University of Maryland. Disponível em: <http://globalequality.info/goi.php?id=87>. Acesso em: 20 jan. 2006.

SOURCEFORGE. *The world's largest Open Source software development web site*. Disponível em: <http://sourceforge.net/>. Acesso em: 8 fev. 2008.

TAURION, C. *Software Livre* – Potencialidades e Modelos de Negócio. Rio de Janeiro: Brasport, 2004.

VARIAN, H. R.; SHAPIRO, C. *A economia da informação*: como os princípios econômicos se aplicam à era da Internet. Campus, 1999.

WATSON, R. T.; BOUDREAU, M.-C.; YORK, P.; GREINER, M.; WYNN, D. *The Business of Open*

Source. Communications of the ACM, 2008. (Forthcoming).

WATSON, R. T.; BOUDREAU, M.-C.; YORK, P.; GREINER, M.; WYNN, D.; GUL, R. Governance and global communities. *Journal of International Management*, n. 11, p. 125-142, 2005.

WHANG, S. Contracting for Software Development. *Management Science*, v. 38, n. 3, p. 307-324, 1992.

WILLIAMSON, O. E. *The economic institutions of capitalism*: firms, markets, relational contracting. New York: Free Press, 1985.

7. BIBLIOGRAFIA COMPLEMENTAR

CRESWELL, J. W. *Research Design*: Qualitative, Quantitative, and Mixed Method Approaches. 2. ed. Thousand Oaks: Sage Publications, 2003.

NETO, C. G.; AUGUSTO, M. P. Um estudo sobre as motivações e orientações de usuários e programadores brasileiros de *software* livre. ENCONTRO DA ASSOCIAÇÃO NACIONAL DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO, 29., 2005, Brasília. *Anais*... Brasília: ANPAD, 2005.

PRICEWATERHOUSECOOPERS. *Lei do Software e seu regulamento*. São Paulo: Editora Atlas S.A, 1999.

RATTNER, H. Inovação tecnológica e pequenas empresas: uma questão de sobrevivência. *Revista de Administração de Empresas*, v. 24, n. 3, p. 70-73, jul.-set. 1984.

RIORDAN, M.; WILLIAMSON, O. Asset Specificity and Economic Organization. *International Journal of Industrial Organization*, v. 3, 1985.

SAPPINGTON, D. Incentives in Principal-Agent Relationships. *Journal of Economic Perspectives*, v. 5, n. 2, p. 45-66, 1991.

TAKAHASHI, T. *Sociedade da Informação no Brasil* – Livro Verde. Ministério da Ciência e Tecnologia, 2000.

WHEELER, D. A. *Why Open Source Software/Free Software (F/LOSS/FS)?* Look at the Numbers! 14 Nov. 2005. Disponível em: <http://www.dwheeler.com/contactme.html>. Acesso em: 21 jan. 2006.